

### REMARKS

The Examiner found the arguments presented in Applicant's Appeal Brief to be persuasive and, as a result, vacated the finality of the last office action. The Examiner, however, now finally rejects claims 25, 27-31, 34-38, 40-41, and 43-47 as being anticipated under 35 U.S.C. 102(b) by U.S. Patent No. 6,263,487 to Stripf et al. ("Stripf"). Reconsideration of the final rejection is respectfully requested in view of the following remarks.

1. **35 U.S.C. 102(b) Rejections**

Claims 25, 27-31, 34-38, 40-41, and 43-47 are now rejected under 35 U.S.C. 102(b) as being anticipated by Stripf.

a. **Independent Claim 25**

First, Applicant notes that independent Claim 25 currently recites:

A method for executing a program for an industrial automation system, comprising:  
providing a computer unit with input aids, output aids and a display device, having modules and functions respectively representing sub-tasks of an automation solution being modeled and/or created using the input aids and optionally the display device, having the modules and functions being structured and networked using the input aids and optionally the display device as to form at least one hierarchical tree as a machine-independent program,

loading the machine-independent program in the form of the at least one hierarchical tree into the corresponding components of the automation system, wherein the corresponding components of the automation system execute the machine-independent program present in the form of the at least one hierarchical tree with the aid of at least one object machine assigned to the corresponding components of the automation system, and wherein the at least one object machine provides operators and objects from which the machine-independent program is provided in the form of the at least one hierarchical tree; and

during or after loading of the machine-independent program, instantiating the operators using the at least one object machine into corresponding components of the automation system; and

converting the symbolic representation of the hierarchical tree to physical addresses to generate a loadable program in the form of an executable program or operator tree.

Per MPEP 2131, "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). In addition, "[t]o anticipate a claim, the identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Applicant submits that Stripf fails to expressly or inherently describe at least "modules and functions being structured and networked using the

input aids and optionally the display device as to form at least one hierarchical tree as a machine-independent program” as required in independent Claim 25.

According to the Examiner, Stripf discloses “software function blocks to be loadable directly into a programmable logic controller over the Internet and the Internet communications interface, so that they can be tied in to the control program while it is running, the software function blocks have an object-oriented design” at col. 1, line 65 to col. 2, lines 25; Fig. 2; and col. 2, lines 44-46 of Stripf. *See* pp. 3-4 of the October 3, 2008 Office Action. The Examiner’s position is that this alleged disclosure of Stripf inherently or expressly describes “modules and functions being structured and networked using the input aids and optionally the display device as to form at least one hierarchical tree as a machine-independent program” as claimed. Applicant respectfully disagrees with the Examiner’s position.

As stated at paragraph [0035] of the Substitute Specification of the present application:

In the context of the invention, a machine-independent program 33 modeled or created on the engineering device 22 preferably via objects 27 is not first converted to a machine-dependent automation program in the form of a sequential machine code and then loaded onto a component of the automation system or another destination device, as is normally the case with the prior art, rather the machine-independent programs are loaded into the corresponding components of the automation system in the form of hierarchical trees. (emphasis added).

First, Stripf does not expressly or inherently describe “modules and functions being structured and networked using the input aids and optionally the display device as to form at least one hierarchical tree as a machine-independent program” as claimed because there is no express or inherent disclosure of a hierarchical tree in Stripf. Instead, Stripf discloses the typical compilation of objects in the form of “sequential machine codes” of the prior art mentioned above at paragraph [0035] of the present application, which is clearly different from the form of the claimed “hierarchical tree.” *See* col. 2, lines 57-59 of Stripf, for example: “On a user level, only one compiler is required for all the programmable controllers to create the machine codes.” (emphasis added). Accordingly, Applicant submits that independent Claim 25 is not anticipated by Stripf because Stripf discloses software function blocks in the form of sequential machine codes, not in the form “at least one hierarchical tree” as claimed. Accordingly, Stripf does not expressly or inherently describe “modules and functions being structured and networked using the input aids and optionally the display device as to form at least one hierarchical tree as a

machine-independent program” as required in independent Claim 25. Accordingly, Applicant respectfully submits that independent Claim 25 is now in condition for allowance.

**b. Independent Claim 38**

For the reasons set forth above with respect to Claim 25, Applicant also submits that Stripf fails to inherently or expressly describe all the elements of Claim 38, namely at least “a component to load the machine-independent program in the form of the at least one hierarchical tree into the corresponding components of the automation system with the corresponding components of the automation system executing the machine-independent program present in the form of the at least one hierarchical tree.” Accordingly, Applicant also respectfully submits that independent Claim 38 is in condition for allowance.

**c. Independent Claim 47**

Moreover, independent Claim 47 requires a computer program implementing a method for executing a program for an industrial automation system, comprising...“modules and functions being structured and networked using the input aids and optionally the display device so as to form a hierarchical tree as a machine-independent program.” For the reasons provided above with respect to independent Claim 25, Applicant also respectfully submits that independent Claim 47 is in condition for allowance.

**d. Dependent Claims**

**ii. Dependent claims 27-31 and 34-37**

Dependent claims 27-31 and 34-37 are dependent on independent Claim 25. Thus, for at least the reasons set forth above with respect to independent Claim 25, Applicant submits dependent claims 27-31 and 34-37 are in condition for allowance.

**ii. Dependent claims 40-41 and 43-46**

In addition, dependent claims 40-41 and 43-46 are dependent on independent Claim 38. Thus, for at least the reasons set forth above with respect to independent Claim 38, Applicant also submits dependent claims 40-41 and 43-46 are in condition for allowance.

**iii. Dependent claims 37 and 46**

Moreover, Applicant submits that dependent claims 37 and 46 provide further reasons for allowance. Claims 37 and 46 each require that the objects of the machine-independent program present as a hierarchical object or operator tree are assigned a collection of infrastructure services

or infrastructure functions that access the objects via containers assigned to the objects such that an infrastructure service or an infrastructure function can be used by all the objects.

The Examiner's position is that Stripf discloses these limitations at col. 2, lines 35-40; Figure 2; and col. 2, lines 44-46 of Stripf. In particular, the Examiner characterizes these locations in Stripf as describing that "the portability of the code ensures that a programmable controller with a[n] execution system in the form of a Java byte code interpreter can process the Java function blocks sent to the programmable controller over the internet independently of a processor hardware architecture of the programmable controller." See pp. 11-12 of the October 3, 2008 Office Action. According to the Examiner, the limitations of dependent claims 37 and 46 are anticipated by this disclosure. Applicant respectfully disagrees with the Examiner's position.

Per MPEP 2131, "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). Stripf fails to do so. There is no express description in Stripf of "containers assigned to the objects such that an infrastructure service or an infrastructure function can be used by all the objects" as required in dependent claims 37 and 46. For this reason alone, dependent Claims 37 and 46 are not anticipated by Stripf.

Further, containers are certainly not an inherent element of the Java function blocks or objects of Stripf, nor has the Examiner provided any evidence of the same. Per MPEP 2112, "[i]n relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art." *Ex parte Levy*, 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990)." Thus, even assuming Stripf discloses that the portability of its code ensures that a programmable controller with an execution system in the form of a Java byte code interpreter can process the Java function blocks sent to the programmable controller over the internet independently of a processor hardware architecture of the programmable controller. nothing in Stripf expressly or inherently describes "containers assigned to the objects such that an infrastructure service or an infrastructure function can be used by all the objects" as claimed in dependent claims 37 and 46. In view of the above, dependent claims 37 and 46 provide further reasons for allowance.


Serial No. 10/560,839  
Atty. Doc. No. 2003P06167WOUS

3. Conclusion

Accordingly, Applicant submits that all claims are in condition for allowance and requests that a Notice of Allowance be issued. The commissioner is hereby authorized to charge any appropriate fees due in connection with this paper, including the fees specified in 37 C.F.R. §§ 1.16 (c), 1.17(a)(1) and 1.20(d), or credit any overpayments to Deposit Account No. 19-2179.

Respectfully submitted,

Dated: 12/3/18

By:   
John P. Musone  
Registration No. 44,961  
(407) 736-6449  
Siemens Corporation  
Intellectual Property Department  
170 Wood Avenue South  
Iselin, New Jersey 08830